

5 **The Claims:**

1. A software engine for application loading a software application onto a user's machine, wherein a core service of the application is loaded onto the user's machine to enable the application to commence to operate on the user's machine, the engine subsequently loading non-core services of the application according
10 to a priority order determined by the engine.

2. A software engine as claimed in claim 1, wherein the engine is part of the core service and is loaded with the core service.

15 3. A software engine as claimed in claim 2, wherein the engine commences operation upon completion of loading of the core service.

20 4. A software engine as claimed in claim 1, wherein the priority order includes a top priority, top priority being given to any non-core service of the application required to be on the user's machine as a result of interaction with the application by the user.

25 5. A software engine as claimed in claim 1, wherein the non-core services are loaded in a manner controlled by the engine.

6. A software engine as claimed in claim 5, wherein the control takes into account user interaction with the application.

30 7. An engine as claimed in claim 5, wherein before loading the non-core services they are registered with the engine.

8. A software engine as claimed in claim 7, wherein the engine checks the registration list of non-core services before loading a requested non-core service.

35 9. A software engine as claimed in claim 1, wherein there is provided a cache into which at least one object for the application can be stored.

5

10. A software engine as claimed in claim 9, wherein the engine includes a memory management module that keeps track of usage of cached objects; the memory management module being able to de-allocate one or more of the objects.

10 11. A software engine as claimed in claim 10, wherein the cache is operative only when the application is on the user's machine.

12. A software engine as claimed in claim 9, wherein the cache includes an object repository into which the at least one object is placed, and an object description.

15

13. A software engine as claimed in claim 12, wherein the object description includes one or more selected from the group consisting of: object reference, object key, reference counter and time stamp.

20

14. A software engine as claimed in claim 10, wherein the de-allocation of one or more of the objects includes an arbitrary time offset.

25

15. A software engine as claimed in claim 14, wherein if the object description of an object in the object repository has a reference counter equal to zero for a time equal to at least the time offset, the corresponding object description is removed from the object repository.

16. A software engine as claimed in claim 1, wherein the loading is downloading over the Internet.

30

17. A method of loading a software application onto a user's machine using a software engine, the method including loading onto the user's machine core services of the application to enable the user to interact with the application, and loading non-core services of the application according to a priority order determined by the engine.

35

5 18. A method as claimed in claim 17, wherein the engine is part of the core service
and is loaded with the core service.

10 19. A method as claimed in claim 18, wherein the engine commences operation
upon completion of loading of the core service.

15 20. A method as claimed in claim 17, wherein before loading the non-core services
they are registered with the engine.

20 21. A method as claimed in claim 17, wherein with the priority order there is created
a top priority, top priority being given to any non-core services of the application
required to be on the user's machine as a result of interaction with the
application by the user.

25 22. A method as claimed 21, wherein upon interaction with the application by the
user, the application requests the engine to load at least one of the non-core
services, the engine checks the registration and gives the at least one non-core
service top priority for loading.

30 23. A method as claimed in claim 21, wherein the loading of the non-core services is
in a manner controlled by the engine.

35 24. A method as claimed in claim 23, wherein the control takes into account user
interaction with the application.

30 25. A method as claimed in claim 17, wherein objects of the application are storable
in a cache for reuse.

35 26. A method as claimed in claim 25, wherein cached objects are tracked using a
memory management module of the engine, which can de-allocate one or more
of the objects.

5 27. A method as claimed in claim 26, wherein the cache is operative only when the application is on the user's machine.

10 28. A method as claimed in claim 25, wherein the objects are placed into an object repository in the cache, together with an object description.

15 29. A method as claimed in claim 28, wherein the object description includes one or more selected from the group consisting of: object reference, object key, reference counter and time stamp.

20 30. A method as claimed in claim 26, wherein de-allocation includes an arbitrary time offset.

25 31. A method as claimed in claim 30, wherein if the object description of an object repository has a reference counter to equal to zero for a time equal to at least the time offset, the corresponding object description is removed from the object repository.

30 32. A method as claimed in claim 17, wherein the loading in downloading over the Internet.

33. A computer memory management system for use with a software application, the memory management system including a cache, and wherein objects of the application are storable in the cache for reuse.

34. A system as claimed in claim 33, wherein the cache is operative only when the application is on the user's machine.

35. A system as claimed in claim 33, wherein the objects are placed into an object repository in the cache, together with an object description.

5 36. A system as claimed in claim 35, wherein the object description includes one or
more selected from the group consisting of: object reference, object key,
reference counter and time stamp.

10 37. A system as claimed in claim 33², wherein cached objects are tracked using a
memory management module, which can de-allocate one or more of the objects.

38. A system as claimed in claim 37, wherein de-allocation includes an arbitrary
time offset.

15 39. A system as claimed in claim 38, wherein if the object description of an object
repository has a reference counter to equal to zero for a time equal to at least the
time offset, the corresponding object description is removed from the object
repository.